DCON Communication Driver

Driver for Serial Communiation with PC / PAC (WinCon, Nuwa, Hero, WinPAC, ViewPAC) and DCON modules (I-7000, I-8000, and I-87K) using DCON Protocol

Contents

1. Introduction

The DCON driver enables communication between Studio system and most of ICP DAS modules from families DCON (I-7000, I-8000 and I-87K), in accordance with the characteristics covered in this document.

This document contains 8 parts as follow:

- **Introduction: Provide an overview of the driver documentation.**
- **General characteristics: Provide information necessary to identify all the required components (hardware and software) necessary to implement the communication and global characteristics about the communication.**
- **Installation: Explain the procedures that must be followed to install the software and hardware required for the communication.**
- **Driver configuration: Provide the required information to configure the communication driver such as the different permutations for configuration and its default values.**
- **Execution: Explain the steps to test whether the driver was correctly installed and configured.**
- **Troubleshooting: Supply a list of the most common error codes for this protocol and the procedures to fix them.**
- **Application Sample: Provide a sample application for testing the configuration the driver.**
- **History of versions: Provide a log of all the modifications done in driver.**

Note: This document presumes that the user has read the chapter Driver Configuration of the Studio's Technical reference manual.

2. General Information

2.1 Device Characteristics

- Manufacturer: ICP DAS Co., Ltd.

- Compatible Equipment
  -ICP DAS modules from families DCON (I-7000, I-8000 and I-87K).

Note: Please refer to section 2.4 to see the Equipment used in the standard conformance tests for this driver.

Note: All analog modules must be configured to engineering units.

Note: This Driver version does not implement the CRC. The equipment must be configured not to use CRC check.

2.2 Link Characteristics

- Device communication port: RS232 port

- Physical protocol: RS232/RS485

- Logic protocol: ASCII (Proprietary)

- Device Runtime software: DCON driver

- Specific PC Board: None

2.3 Driver Characteristics

- Operation System:

  - Windows XP
  - Windows 2000
  - Windows NT
  - Windows CE (WinCon, Nuwa, Hero, WinPAC, and ViewPAC)

Note:
Please refer to section 2.4 to see the Operating System used in the conformance tests for this driver.

The driver is composed of the following files:

- DCON.INI: Internal files of the driver, it should not be modified by the user.
- DCON.MSG: This file contains the error messages for each error code. It is an internal file of the driver, and the user should not modify it.
- DCON.PDF: This document provides detailed documentation about the driver.
- DCON.DLL: This is the compiled library for the driver.

Note:
All the files above must be in the subdirectory /DRV of the Studio's installation direction.

2.4 Information About Conformance Testing

- Equipment:

Family I-7000: I-7060, I-7021, and I-7012D
Family I-8000: I-8040, I-8041, I-8024, I-8017 and I-8094
Family I-87K: I-87054, I-87017 and I-87024

Configuration:

PLC program:
Baud Rate: 115200
Data Bits: 8
Stop Bits: 1
Parity: None
COM port: 1
Timeout: 200
Checksum: 0 (No Checksum)

Cable:
Operating System (development): NT/2000/XP
Operation System (target): NT/2000/XP
Studio Version: 6.1
Driver version: 1.40

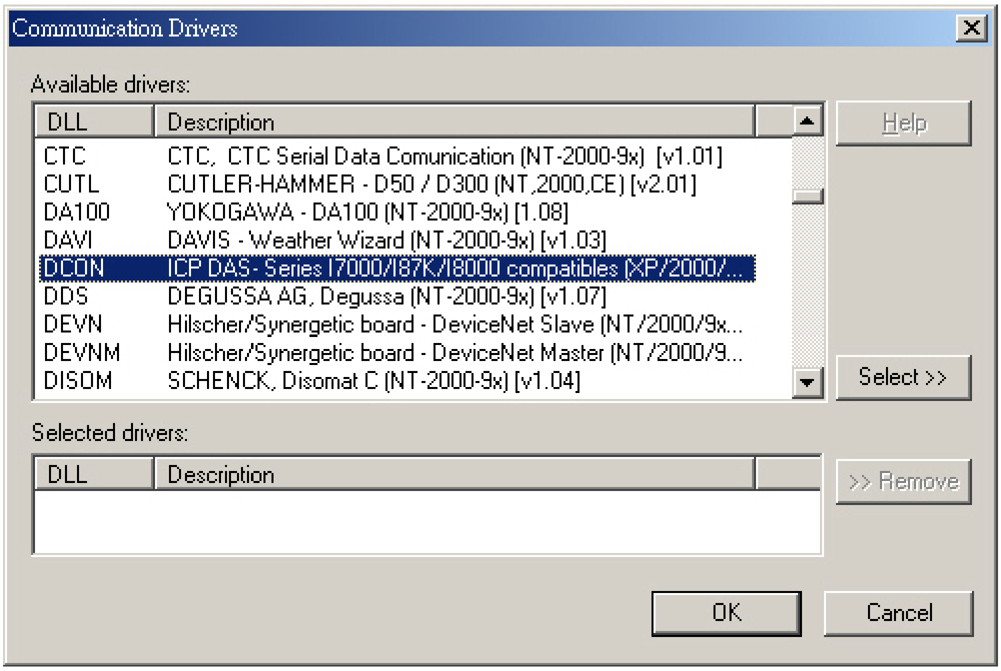
Note:
Users can free download the test demo from the web site: ***http://www.icpdas.com/products/Software/InduSoft/download_iws60.htm***

3. Installation

3.1 Installing the Driver

After install the InduSoft Web Studio (IWS) v3.0 or later version, **the DCON driver of ICPDAS is not installed in IWS**. So before using the DCON driver in IWS, users need to install the DCON driver first. The DCON driver setup file can be downloaded from ICPDAS web site: **http://www.icpdas.com/products/Software/InduSoft/download_iws60.htm**. When the DCON communication driver is installed in IWS, users just need to select the driver in the application and the steps are described as follows:

1. Execute the Studio and select the proper application.
2. Select the menu Insert + Driver….
3. In the column Available Driver, select the **DCON** Driver and push the button **Select>>** (the driver **DCON** must appear in the column **Selected Drivers**)
4. Press **OK**.



Available drivers:

DLL	Description
CTC	CTC, CTC Serial Data Comunication (NT-2000-9x) [v1.01]
CUTL	CUTLER-HAMMER - D50 / D300 (NT,2000,CE) [v2.01]
DA100	YOKOGAWA - DA100 (NT-2000-9x) [1.08]
DAVI	DAVIS - Weather Wizard (NT-2000-9x) [v1.03]
DCON	ICP DAS - Series I7000/I87K/I8000 compatibles (XP/2000/...
DDS	DEGUSSA AG, Degussa (NT-2000-9x) [v1.07]
DEVN	Hilscher/Synergetic board - DeviceNet Slave (NT/2000/9x...
DEVNM	Hilscher/Synergetic board - DeviceNet Master (NT/2000/9...
DISOM	SCHENCK, Disomat C (NT-2000-9x) [v1.04]

Selected drivers:

DLL	Description

3.2 Other Software Requirements

It is not necessary to install any other software in the PC to enable the communication between the host and the Device.

Note:
Special cautions must be taken when installing the physical hardware. Refer to the hardware manufacture documentation for specific instructions in this area.

4. Driver Configuration

After the driver is installed and selected in the Studio (see section 3.1), you should proceed to the driver configuration.

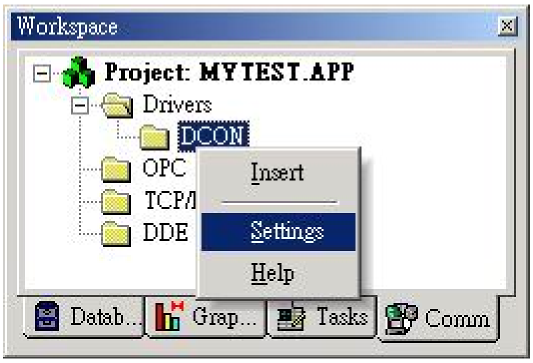
The driver configuration is two parts:

The Settings or Communication parameters, they are only one configuration to the whole driver, and you have the communication tables or Driver Worksheets, where the communication tags are defined.

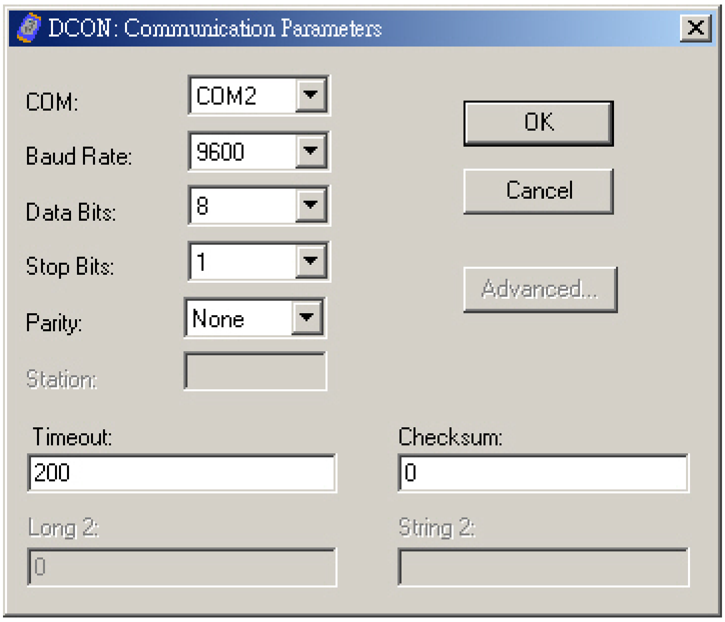
4.1 Other Software Requirements

These parameters are valid for all driver worksheets configured in the system. To open the window for configuring the **Communication parameters**, and follows these steps:

1. In the **Workspace** of the Studio environment, select the **Comm** table.
2. Expand the folder **Driver** and select the subfolder **DCON**.
3. Right click on the **DCON** subfolder and select the option **Settings**.



When selecting the Settings, these are the following dialog to configure:



COM: COM2
Baud Rate: 9600
Data Bits: 8
Stop Bits: 1
Parity: None
Station:
Timeout: 200
Checksum: 0
Long 2: 0
String 2:

4. Specify the parameters as noted in the following table:

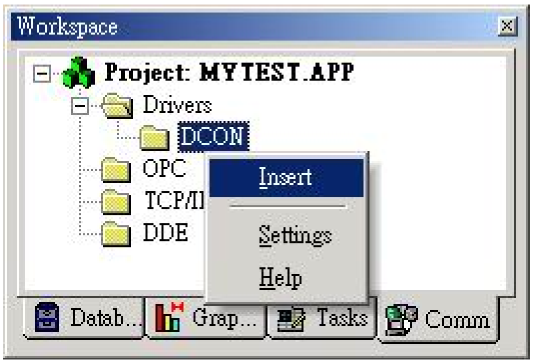
Parameter	Default Value	Valid values	Description
COM	COM2	COM1 to COM256	Serial port of the PC/PAC used to communication with the device
Baud Rate	9600	110 to 115200bps	Communication rate of data
Data Bits	8	5 to 8	Number of data bits used in the protocol
Stop Bits	1	1 or 2	Number of stop bits used in the protocol
Parity	None	Even, odd, none, space or mark	Parity of the protocol
Timeout	200	0~2147483647	Time unit is 1 ms.
Checksum	0	0 or 1	**0:** No checksum **1:** Checksum

Note: These Parameters must be just the same as the configured on the ICP DAS device.

4.2 Driver Worksheet

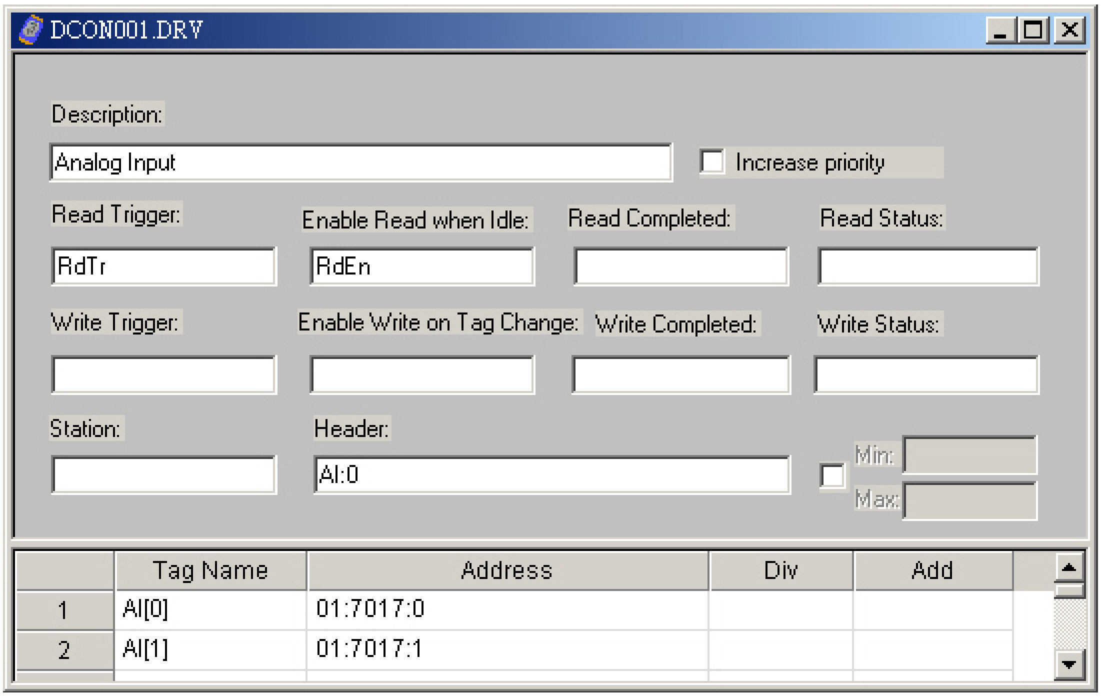
It is possible to configure main driver worksheets and each one will be composed of a Header and Body. To create a new driver worksheet, and follows these steps:

1. In the **Workspace** of the Studio environment, select the table **Comm**.
2. Expand the folder **Drivers** and select the subfolder **DCON**.
3. Right click on the **DCON** subfolder and select the option **Insert**.



Note: To optimize communication and ensure better performance for the system, it is important to tie the tags in different driver sheets according to the events that must trigger the communication of each group of tags and the periodicity for which each group of tags must be written or read. In addition, it is recommended to configure the addresses of communication in sequential blocks.

When creating a communication table, you have the following window:



Description: Analog Input

□ Increase priority

Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:
RdTr	RdEn		
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:
Station:	Header:		
	AI:0	□	

	Tag Name	Address	Div	Add
1	AI[0]	01:7017:0		
2	AI[1]	01:7017:1		

Caution:
Users can only apply the tag name up to 100 items in the same worksheet

All entries at the Driver Worksheet, exception by the Station, Header and Address are standard to all communication drivers. You should refer to Studio Communication Driver documentation about the configuration of the standard fields. This document describes the Station, Header and Address fields, which are specific to each communication driver.

4.3 Station and Header configuration

Parameter	Default Value	Valid value	Description
Station			Not used
Header	AI	See next table	Define the type of variable to be read or written from or to the device.

The Header field defines the type of variables that will be read or written from or to the device. It complies with the syntax:<Module's Type>. After editing the field Header, the system will check if it is valid or not. If the syntax was incorrect, the default value (AI) will be automatically placed in the field.

You can type Tag between curl brackets into this field, but be sure that the Tag's value is correct, with the right syntax, or you will get the Invalid Header error.

Information regarding the parameter "Header"			
Type	**Sample of syntax (Header)**	**Valid range of initial Address**	**Comment**
Read Digital Input/Output	DI	Any	
Read/Write Digital Output	DO	Any	
Read Analog	AI	Any	The ICP DAS device channel must be configured to engineering units
Write/Read Analog	AO	Any	The ICP DAS device channel must be configured to engineering units
Read/Set Counter	Counter	Any	
Read/Set DI Counter value	DICounter	Any	
Write Command	SendCmd	**Command String**	Send command to the device

Note: Always creates two different driver worksheets for Input and Output modules.

4.4 Address Configuration

The body of the driver worksheet allows you to associate each tag to its respective address in the device. In the column **Tag Name**, you must type the tag from your application database. This tag will receive or send values from or to the device.

The address cells complies to the following syntax:

For **I-7000** and **I-87K** Family:

<Module's Address>:<Module ID>:<Channel Number>

Module's Address[Hex]: Module's Address in the network **(Range from 00 to FF)**
Module ID: Module's ID in the device
Channel Number: Channel's Number to be read/written from/to the module.

Sample of Addressing Configuration		
Address on the Device	**Header Field**	**Address Field**
Read I-87053 module Address 1 channel 0 (input)	**DI**	**01:87053:0**
Write/Read I-7063D module Address 6 channel 1 (output)	**DO**	**06:7063:1**
Read I-7017R module Address 3 channel 6 (input)	**AI**	**03:7017:6**
Write/Read I-7021 module Address 0x0A channel 1 (output)	**AO**	**0A:7021:0**
Read/Set I-7080 module Address 7 channel 0 (input)	**Counter**	**07:7080:0**
Read/Set I-7060D DI module counter Address 5 channel 0 (input)	**DICounter**	**05:7060:0**
Write command to devices from com port of PC	**SendCmd**	**Command String (like $01M)**

For I-8000 Family:

<Module's Address>:<Module ID>:<Slot Number >:<Channel Number>

Module's Address[Hex]: Module's Address in the network**(Range from 00 to FF)**
Module ID: Module's ID in the device
Slot Number: The slot number that the Module is plugged in the backplane.
Channel Number: Channel's Number to be read/written from/to the module.

Sample of Addressing Configuration		
Address on the Device	**Header Field**	**Address Field**
Read address 3 channel 0 of I-8053 module in slot 1	**DI**	**03:8053:1:0**
Write DO to address 6 channel 2 of I-8064 module in slot 4	**DO**	**06:8064:4:2**
Read AI from address 2 channel 5 of I-8017H module in slot 6	**AI**	**02:8017:6:5**
Write AO to address 0x0A channel 3 of I-8024 module in slot 0	**AO**	**0A:8024:0:3**

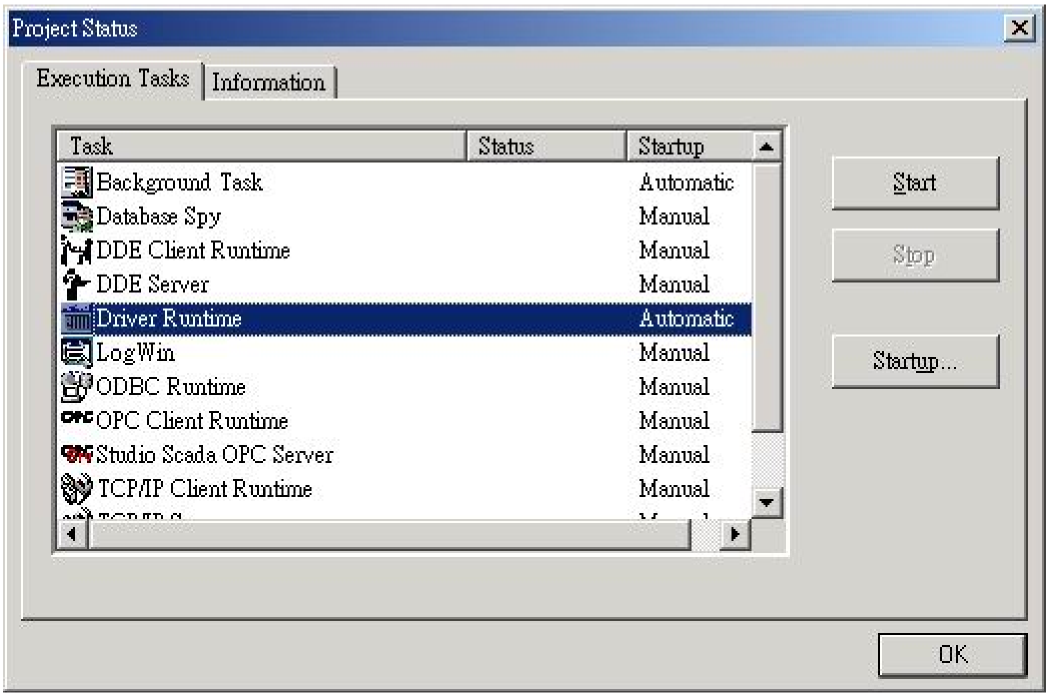
Head Type	Support Modules
DI:	I-7011/12/14/16, I-7041/44/50/52/53/55, I-7060/63/65, I-8040/42/50/51/52/53/54/55/58/63 I-87040/87051/52/53/54/55/58/63/69
DO:	I-7005/11/12/14/16, I-7042/43/44/50, I-7060/63/65/66/67/80, I-8041/42/50/54/55/56/57/60/64/68, I-8060/63/64/65/66/68, I-87041/54/55/56/57 and 63/64/65/66/68
AI:	I-7005/11/12/13/14/15/16/17/18/19/33, 8017H/, 87013/15/16/17/18/19
AO:	I-7016/I-7021/I-7022/I-7024, I-8024, 87022/24/26
Counter:	I-7080
DI Counter	I-7011/12/14/16, I-7041/44/50/52/53/55, I-7060/63/65 I-87040/51/52/53/54/55/63

4.5 Device Configuration

The device parameter (baud rate, stop bits, etc) must be matched with the settings configured in the Communication Parameters of the DCON driver.

5. Execution

When installing the driver, it is automatically selected to execute when you start-up the Runtime Environment. To verify if the driver is correctly enabled and started, use the menu option **Project + Status**…., and verify the task Driver Runtime.



Task	Status	Startup
Background Task		Automatic
Database Spy		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
LogWin		Manual
ODBC Runtime		Manual
OPC Client Runtime		Manual
Studio Scada OPC Server		Manual
TCP/IP Client Runtime		Manual

6. Application in PAC (WinCE) Platform

The PAC (Programmable Automation Controller) is a leading edge embedded platform in Windows CE.NET with the Intel Strong ARM CPU. It can operate as an intelligent distributed data acquisition front end connected to Host machine running a standard SCADA package. The users can develop SCADA applications in a PC, and then download and apply them on the PAC InduSoft Embedded Controller. The application procedure is the same as the description of the above section. Here, we will explore the design steps for PAC only. If users need more detailed information about how to build and design a project, please refer to the above section or InduSoft Web studio user manual.

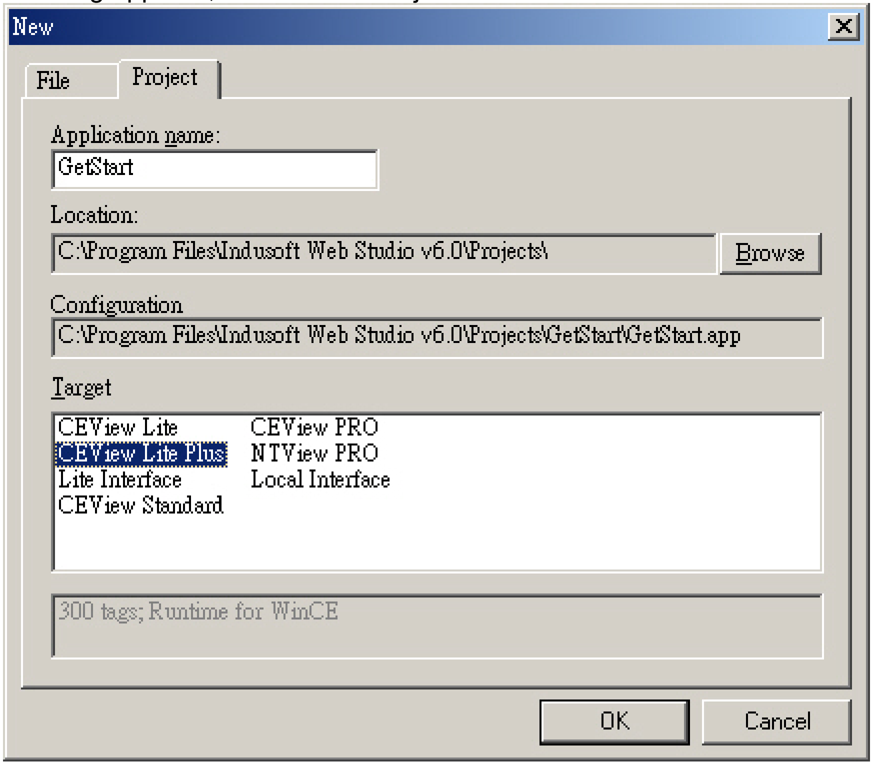
6.1 Creating a New Application for PAC

Users must choose Windows CE target system in creating a New Project. There are many kinds of target systems in InduSoft for you as following table:

Target System	Product License
CEView Lite	150 tags; Runtime for WinCE
CEView Lite Plus	300 tags; Runtime for WinCE
CEView Standard	1500 tags; Runtime for WinCE
CEView PRO	4000 tags; Runtime for WinCE

Follow the procedure to create a new IWS application:

1. Select **File→New** from the InduSoft Web Studio main menu bar.
2. When the New dialog appears, click on the “Project” tab.



Application name: GetStart

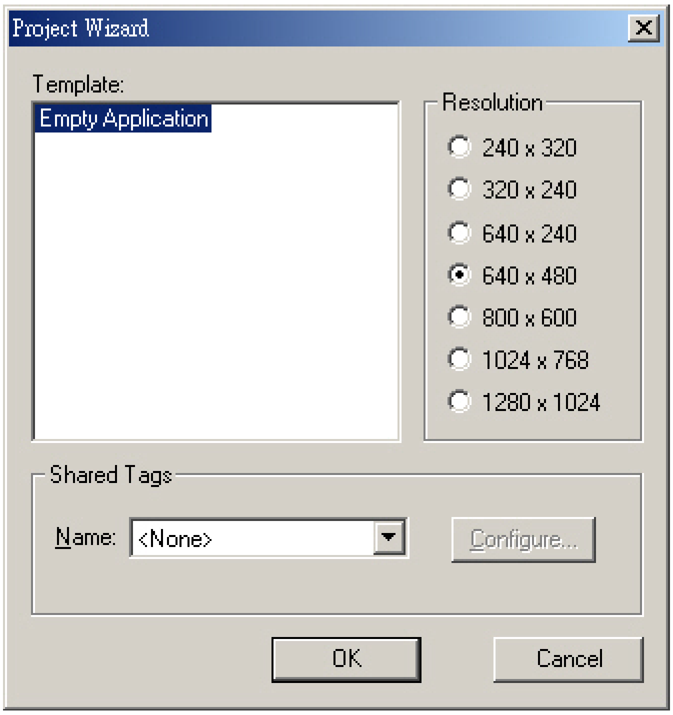
Location: C:\Program Files\Indusoft Web Studio v6.0\Projects\

Configuration: C:\Program Files\Indusoft Web Studio v6.0\Projects\GetStart\GetStart.app

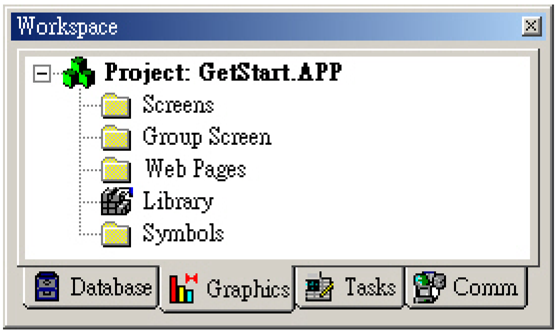
Target: CEView Lite, CEView Lite Plus, Lite Interface, CEView Standard, CEView PRO, NTView PRO, Local Interface

300 tags; Runtime for WinCE

3. Set the name of your application in "Application" field (For this example, we set "**GetStart**"). IWS would automatically create a new directory of the same name and assign your application file to that directory (Refer to the "Configuration" field in the previous figure). To put your application file other than in the \Projects subfolder of InduSoft Web Studio, and click Browse to navigate to the preferred directory location.
4. Select a platform from the Target Platform list (For this example, select "**CEView Lite Plus**"), and click OK to proceed.
5. When the Project Wizard dialog displays, select Empty Application from the Template list. Select the "640 X 480" to specify the application resolution, and click "OK".



The new application file name is shown in the Workspace window as follows.
(For this example, the file name is Project: GetStart.APP).



6.2 Driver Configuration

The PAC can communicate with ICP DAS I-8000 and I-87K series modules across extension slots of the backplane. And it can also be applied to control the distributive controller and ICP DAS DCON (I-7000/I-8000/I-87K) series modules through RS-485 network. Notice that these **Communication parameters** are valid for DCON driver worksheets configured in the PAC COM port. Users can set **Communication parameters** in configuration window, refer to section 4.1.

Note: If these modules are plugged in the PAC slots, don't care the Communication Parameters. But, when you apply remote modules, it is important to set the Communication Parameters. Please check the hardware manual of PAC for COM port assignment.

6.2.1 I-8000 and I-87K Modules Plugged in PAC Extension Slot

The following commands are supported to the modules plugged in slots of the PAC.

Head Type	Support Modules
DI	I-8040/42/50/51/52//53/54/55/58/63/77 I-87040/51/52/53/54/55/58/63
DO	I-8037/41/42/50/54/55/56/57, I-8060/63/64/65/66/68/69/77, I-87041/54/55/57 and I-87063/64/65/66/68/69
AI	I-8017H, I-87013/15/16/17/18/19
AO	I-8024, I-87022/24/26
AI:x	I-8017H
AO:x	I-8024
DI Counter	I-87040/51/52/53/54/55/63
Encoder:x:x:x	I-8090

Note: This x symbol on the I-8017H/I-8024 can be configured to an input/output range for AI/AO in the PAC's slot. The following description lists all the x values. For I-8024, the value 0~3 will be configured as +/-10V. If users just use AI or AO head type without x value, the default x value is 0.

0:	**+/- 10V**
1:	**+/- 5V**
2:	**+/- 2.5V**
3:	**+/- 1.25V**
4:	**+/- 20mA**

This three x symbols on the I-8090 can be configured as X-axis, Y-axis and Z-axis counter mode respectively. The x value of configuration is as follows:

0: ENC_QUADRANT
1: ENC_CW_CCW
2: ENC_PULSE_DIR

Before starting configuring ICP DAS PAC driver, you need to select the driver at the application where it will be used. In the workspace, right click the folder “Drivers” on “Comm” tab to add or remove the communication drivers. On the dialog box opened, choose the DCON driver, and hit “Select”. For more information about driver configuration, refer to section 4.

The body of the driver worksheet allows you to associate each tag to its respective address in the device. In the “**Tag Name”** column, you can set the tag from your application database. Hence, we can exchange data through the tag with the device.

The address cells complies to the following syntax:

For **I-8000** and **I-87K** Family :

<Module ID>:<Slot Number >:<Channel Number>

Module ID: Module’s ID of the device
Slot Number: The slot number that the module has been plugged in the backplane. The WinCON starts from slot 1. The WinPAC and ViewPAC starts from slot 0.
Channel Number: Channel’s Number to be read or written from the module. (If channel 1 of the I-8090 modules is X-axis, channel 2 is Y-axis, and channel 3 is Z-axis.)

Sample of Addressing Configuration		
Address on the Device	**Header Field**	**Address Field**
Read DI from channel 0 of I-8053 module in slot 1 of W-8x3x	DI	8053:1:0
Write DO to channel 1 of I-8064 module in slot 0 of WinPAC	DO	8064:0:1
Read AI from channel 5 of I-87017 module in slot 6 of W-873x	AI	87017:6:5
Write AO (Voltage) to channel 3 of 8024 module in slot 1 of W-8x3x	AO	8024:1:3
Write AO (Current) to channel 2 of 8024 module in slot 0 of ViewPAC	AO:4	8024:0:2
Read AI (Voltage) from channel 3 of I- 8017H module in slot 1 of W-8x3x	AI	8017:1:3
Read AI (Current) from channel 3 of I- 8017H module in slot 1 of W-8x3x embedded controller	AI:4	8017:1:3
Read X-axis CW/CCW counting value from 8090 module in slot 1 of W-8x3x	Encoder:1:1:2	8090:1:1

Read Y-axis Quadrant counting value from 8090 module in slot 1 of W-8x3x	Encoder:2:0:1	8090:1:2
Read Z-axis Pulse/Dir counting value from 8090 module in slot 1 of W-8x3x	Encoder:0:1:2	8090:1:3
String Command to 87k modules in slot 3 of WinCon	SendCmd:3	String Command (like $00M)

6.2.2 Distributive Controller of I-8000, I-87K and I-7000 Series Modules Through COM Port

The following commands are supported to the modules controlled by the PAC COM port through RS-485 network.

Head Type	Support Modules
DI:	I-7011/12/14/15/16, I-7041/44/50/52/53/55, I-7060/63/65, I-8040/42/50/51/52//53/54/55/58/63/77 and I-87040/51/52/53/54/55/58/63
DO:	I-7005/11/12/14/16, I-7042/43/44/50, I-7060/63/65/66/67/80, I-8037/41/42/50/54/55/56/57, I-8060/63/64/65/66/68/69/77, I-87041/54/55/57 and I-87063/64/65/66/68/69
AI:	I-7005/11/12/13/14/15/16/17/18/19/33, I-8017H, and I-87013/15/16/17/18/19
AO:	I-7016/2122/24, I-8024, and I-87022/24/26
DI Counter:	I-7011/12/14/16, I-7041/44/50/52/53/55, I-7060/63/65 and I-87040/51/52/53/54/55/63

If you want to use PAC Com port to connect to ICP DAS I-7000/I-8000/I-87K series modules, they are in conformity with PC COM port connection. The address cells complies to the following syntax:

For **I-7000** and **I-87K** Family:

<Module's Address>:<Module ID>:<Channel Number>

Module's Address[Hex]: Module's Address in the network **(Range from 00 to FF)**
Module ID: Module's ID in the device
Channel Number: Channel's Number to be read/written from/to the module.

Sample of Addressing Configuration		
Address on the Device	**Header Field**	**Address Field**
Read I-87053 module Address 1 channel 0 (input)	DI	01:87053:0
Write/Read I-7063D module Address 6 channel 1 (output)	DO	06:7063:1
Read I-7017C module Address 3 channel 6 (input)	AI	03:7017:6

Write/Read I-7021P module Address 0x0A channel 1 (output)	AO	0A:7021:1
Read/Set I-7080 module Address 1 channel 0 (input)	Counter	03:7080:0
Read/Set I-7060 DI module counter Address 5 channel 1 (input)	DICounter	05:7060:1
String Command to devices from com port of WinCon	SendCmd	String Command (like $00M)

For I-8000 Family:

<Module's Address>:<Module ID>:<Slot Number >:<Channel Number>

Module's Address[Hex]: Module's Address in the network**(Range from 00 to FF)**
Module ID: Module's ID in the device
Slot Number: The slot number that the Module is plugged in the backplane.
Channel Number: Channel's Number to be read/written from/to the module.

Sample of Addressing Configuration		
Address on the Device	**Header Field**	**Address Field**
Read address 2 channel 0 of I-8053 module in slot 1	DI	02:8053:1:0
Write DO to address 6 channel 3 of I-8064 module in slot 4	DO	06:8064:4:3
Read AI from address 3 channel 5 of I-8017H module in slot 6	AI	03:8017:6:5
Write AO to address 0x0A channel 4 of I-8024 module in slot 1	AO	0A:8024:1:4

6.2.3 Configuring I-8094/I-8094F Motion Control Modules (WinPAC and ViewPAC only)

The following commands are supported to the I-8094 and I-8094F motion control modules.

Head Type	Support Modules
I8094	I-8094/I-8094F

For Writing Command to I-8094/I-8094F:

- **Tag Name Field**

The body of the driver worksheet allows you to associate each tag to its respective I-8094 API function. In the column **Tag Name**, you must type the tag from your application database. This tag must be configured as **String Type** and will send its value to an API function according to Address field defined.

The tag value has the following sequence:

<Variable 1>:< Variable 2>:< Variable 3>: ・・・ < Variable N>

Variable 1 represents the first parameter of the API been called. The variable 2 represents second parameter of the API and so on. Parameters are used with **Decimal** value.

Example of Tag Value	
I-8094 Function	**Tag Value(String Type)**
i8094MF_SET_PULSE_MODE( ); **Card No: 1 axis: 1 (0x1 → X) mode: 5 (Pulse)**	**1:1:5**
i8094MF_SET_MAX_V( ); **Card No: 2 axis: 15 (0xf → XYZU) data: 2000000**	**2:15:2000000**

■ Address Field

To define the API function to be called, please fill in the function type ID lists below in address field. IDs are used with **Hexadecimal** value.

Type ID and Function Reference Table (Write)			
Type ID	**Function**	**Type ID**	**Function**
01	i8094MF_REGISTRATION();	**0D**	i8094MF_CLEAR_SLMT();
03	i8094MF_SET_PULSE_MODE();	**0E**	i8094MF_CLEAR_S();
04	i8094MF_SET_MAX_V();	**0F**	i8094MF_SET_ALARM();
05	i8094MF_SERVO_ON();	**10**	i8094MF_SET_INPOS();
06	i8094MF_SERVO_OFF();	**11**	i8094MF_SET_FILTER();
07	i8094MF_RESET_CARD();	**12**	i8094MF_VRING_ENABLE();
08	i8094MF_SET_HLMT();	**13**	i8094MF_VRING_DISABLE();
09	i8094MF_LIMITSTOP_MODE();	**14**	i8094MF_AVTRI_ENABLE();
0A	i8094MF_SET_NHOME();	**15**	i8094MF_AVTRI_DISABLE();
0B	i8094MF_SET_HOME_EDGE();	**16**	i8094MF_SET_LP();
0C	i8094MF_SET_SLMT();	**18**	i8094MF_SET_EP();
Type ID	**Function**	**Type ID**	**Function**
1F	i8094MF_CLEAR_STOP();	**3E**	i8094MF_VSTOP_SUDDENLY();
21	i8094MF_FRNET_SA();	**3F**	i8094MF_INTP_END();
22	i8094MF_NORMAL_SPEED();	**40**	i8094MF_LINE_2D();
23	i8094MF_SET_SV();	**41**	i8094MF_LINE_3D();
24	i8094MF_SET_V();	**42**	i8094MF_ARC_CW();

25	i8094MF_SET_A();	**43**	i8094MF_ARC_CCW();
26	i8094MF_SET_D();	**44**	i8094MF_CIRCLE_CW();
27	i8094MF_SET_K();	**45**	i8094MF_CIRCLE_CCW();
28	i8094MF_SET_L();	**46**	i8094MF_EXD_MP();
29	i8094MF_SET_AO();	**47**	i8094MF_EXD_FP();
2A	i8094MF_STOP_SLOWLY();	**48**	i8094MF_EXD_CP();
2B	i8094MF_STOP_SUDDENLY();	**49**	i8094MF_EXD_DISABLE();
2C	i8094MF_SET_HV();	**4A**	i8094MF_RECTANGLE();
2D	i8094MF_HOME_LIMIT();	**4B**	i8094MF_LINE_2D_INITIAL();
2E	i8094MF_SET_HOME_MODE();	**4C**	i8094MF_LINE_2D_CONTINUE();
2F	i8094MF_HOME_START();	**4D**	i8094MF_LINE_3D_INITIAL();
31	i8094MF_FIXED_MOVE();	**4E**	i8094MF_LINE_3D_CONTINUE();
32	i8094MF_CONTINUE_MOVE();	**4F**	i8094MF_MIX_2D_INITIAL();
34	i8094MF_AXIS_ASSIGN();	**50**	i8094MF_CONTINUE_INTP();
35	i8094MF_VECTOR_SPEED();	**52**	i8094MF_HELIX_3D();
36	i8094MF_SET_VSV();	**53**	i8094MF_RATIO_INITIAL();
37	i8094MF_SET_VV();	**54**	i8094MF_RATIO_2D();
38	i8094MF_SET_VA();	**55**	i8094MF_SYNC_ACTION();
39	i8094MF_SET_VD();	**56**	i8094MF_SET_COMPARE();
3A	i8094MF_SET_VK();	**58**	i8094MF_SET_PRESET();
3B	i8094MF_SET_VL();	**59**	i8094MF_HELIX_3D();
3C	i8094MF_SET_VAO();	**5A**	i8094MF_RATIO_INITIAL();
3D	i8094MF_VSTOP_SLOWLY();		

For detail information about API functions, please refer to user manual downloaded
from ftp://ftp.icpdas.com/pub/cd/winconcd/napdos/wince/i8094_for_wincon/doc/i8094_manual_1.5tc.pdf

For Receiving Data from I-8094/I-8094F:

■ Tag Name Field

In the column **Tag Name**, you must type the tag from your application database. This tag will receive the data values from the I-8094/I-8094F.

■ Address Field

Please fill column **Address** with the following sequence:

<Function Type ID>:<Variable 1>:<Variable 1>: · · · < Variable N>

The **Function Type ID** with **Hexadecimal** value uses for retrieving data from I-8094/I-8094F is listed below. **Variable 1** represents the first parameter of the API been called. The variable 2 represents second parameter of the API and so on. Parameters are used with **Decimal** value.

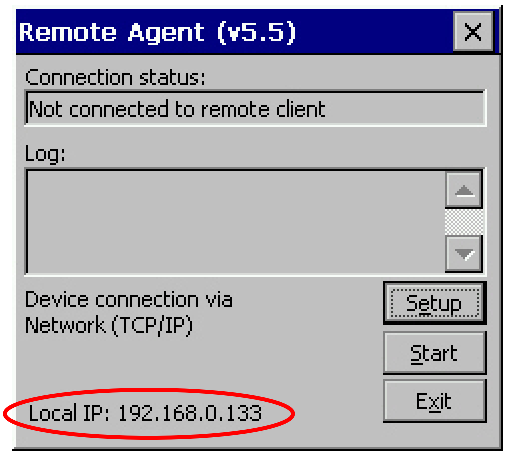
Function Type ID Reference Table (Read)			
Type ID	**Function**	**Type ID**	**Function**
02	i8094MF_GET_VERSION();	**1D**	i8094MF_GET_ERROR();
17	i8094MF_GET_LP();	**1E**	i8094MF_GET_ERROR_CODE();
19	i8094MF_GET_EP();	**20**	i8094MF_FRNET_RA();
1A	i8094MF_GET_CV();	**30**	i8094MF_HOME_WAIT();
1B	i8094MF_GET_CA();	**33**	i8094MF_STOP_WAIT();
1C	i8094MF_GET_DI();	**57**	i8094MF_GET_LATCH();

Example of Addressing Configuration		
Slot Modules (I-8094)	**Header Field**	**Address Field**
Get axis' logic position **Card Number : 5 / Axis : 4 (Z)**	**I8094**	**17:5:4**

6.3 Download Application into PAC

The PAC embedded controller provides two communication media for application with PC design platform. The first method is to download the application through Ethernet communication by connection with the Ethernet port of PAC and PC development platform. ICP DAS recommends using this TCP/IP communication because of performance reasons. Secondly, you can use the COM2 port (RS-232) of WinCon embedded controller to connect to COM1/COM2 port of PC development platform. After completing the application, you can download the application to remote PAC embedded controller that is running CEView on Windows CE. The following steps will show you how to download an application to Compact Flash memory of the W-8x36/W-8x39 or Micro SD memory of the WinPAC/ViewPAC:

1. In PAC embedded controller, double click **\CompactFlash\InduSoft\CEServer.exe** or **\Micro_SD\InduSoft\CEServer.exe** file to verify that the Remote Agent (CEServer.exe) is running.



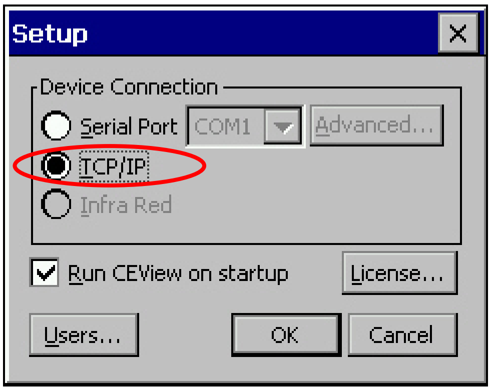
Connection status: Not connected to remote client

Device connection via Network (TCP/IP)

Local IP: 192.168.0.133

2. When the Remote Agent dialog displays, click the Setup button.

3. When the Setup dialog displays, click Serial or TCP/IP to specify how the device is connected to the development station.

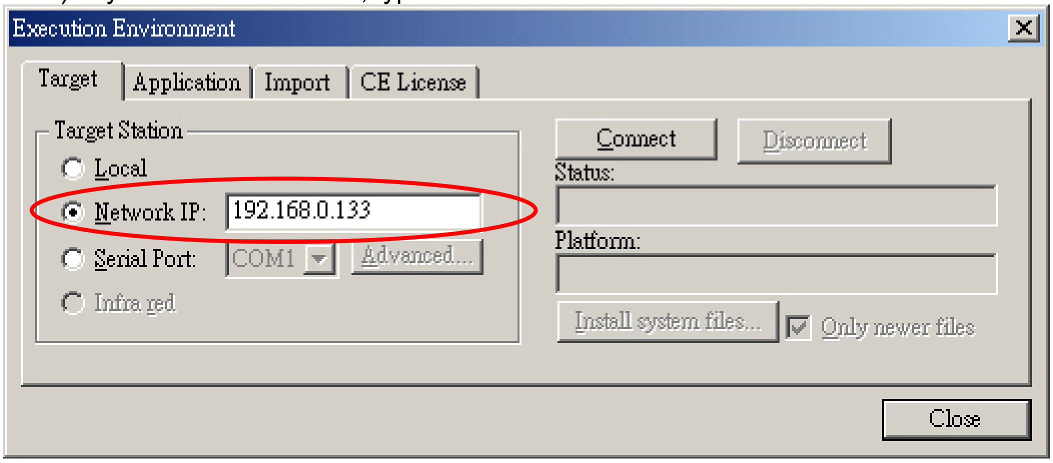


○ Serial Port COM1
● TCP/IP
○ Infra Red
☑ Run CEView on startup

4. Click OK to start running the Remote CEServer in PAC.

5. Select Project→Execution Environment from the main menu bar in the development station to open the Execution Environment dialog. Notice that this must be followed in PC development platform.

6. Specify the communication link type between the development station and target Station (Network IP or Serial Port). If you select Network IP, type the remote station IP Address into the text box.



○ Local
● Network IP: 192.168.0.133
○ Serial Port: COM1
○ Infra red
☑ Only newer files

7. Click the Connect button to connect to the remote PAC.

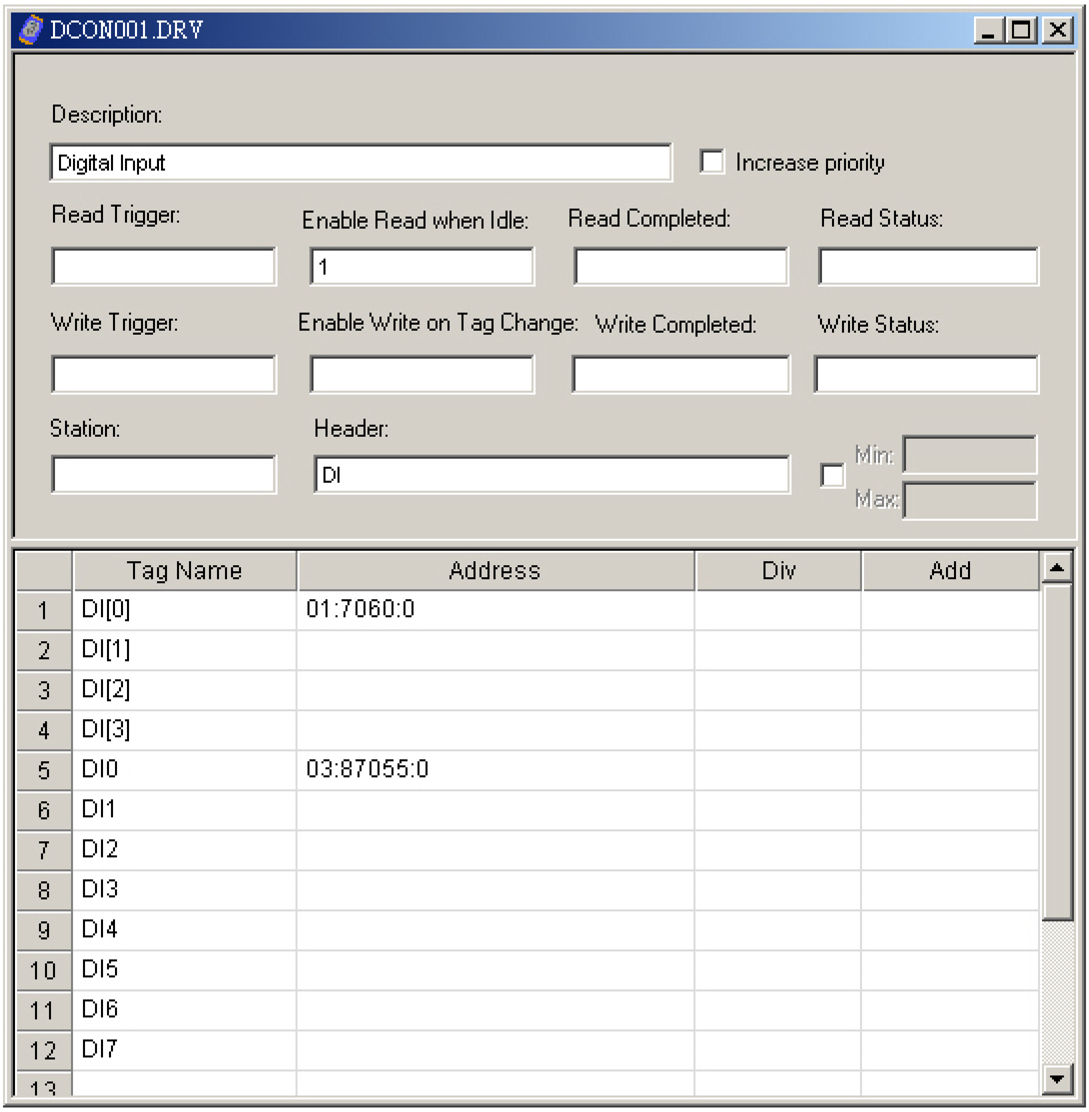
Note: If the remote PAC does not have CEView runtime files, click the Install system files button to download the CEView runtime files to the remote PAC.

8. Click “Send to Target” to download the application to the remote PAC in “Application” tab.

9. After all of the application files are downloaded, click “Run” button to execute the application in the remote PAC embedded controller.

7. Improve DI/AI scan time

The DCON driver version 1.10 provides a simple method to improve multi-channel scan time for a DI/AI module. If a module has many channels, this method can decrease the communication times to obtain all channels values from this module. Users only need to configure the address for the first channel of this module, and give the others null address as Figure 7-1. For example, The I-7060 has 4 digital-input channels and the I-87055 has 8 digital-input channels. Traditionally, users need to use the method as Figure 7-2 to get the each channel value. Therefore, this method needs to communicate with I-7060 for 4 times, and need to communicate with I-87055 for 8 times. By using the method shown in Figure 7-1, users just need to communicate with the I-7060 and I-87055 once respectively. Then, the all channels values will be obtained for these two modules. So **we suggest to use "Fast Mode" (like Figure 7-1) to speed up the module communication**.



Description: Digital Input

□ Increase priority

Read Trigger:

Enable Read when Idle: 1

Read Completed:

Read Status:

Write Trigger:

Enable Write on Tag Change:

Write Completed:

Write Status:

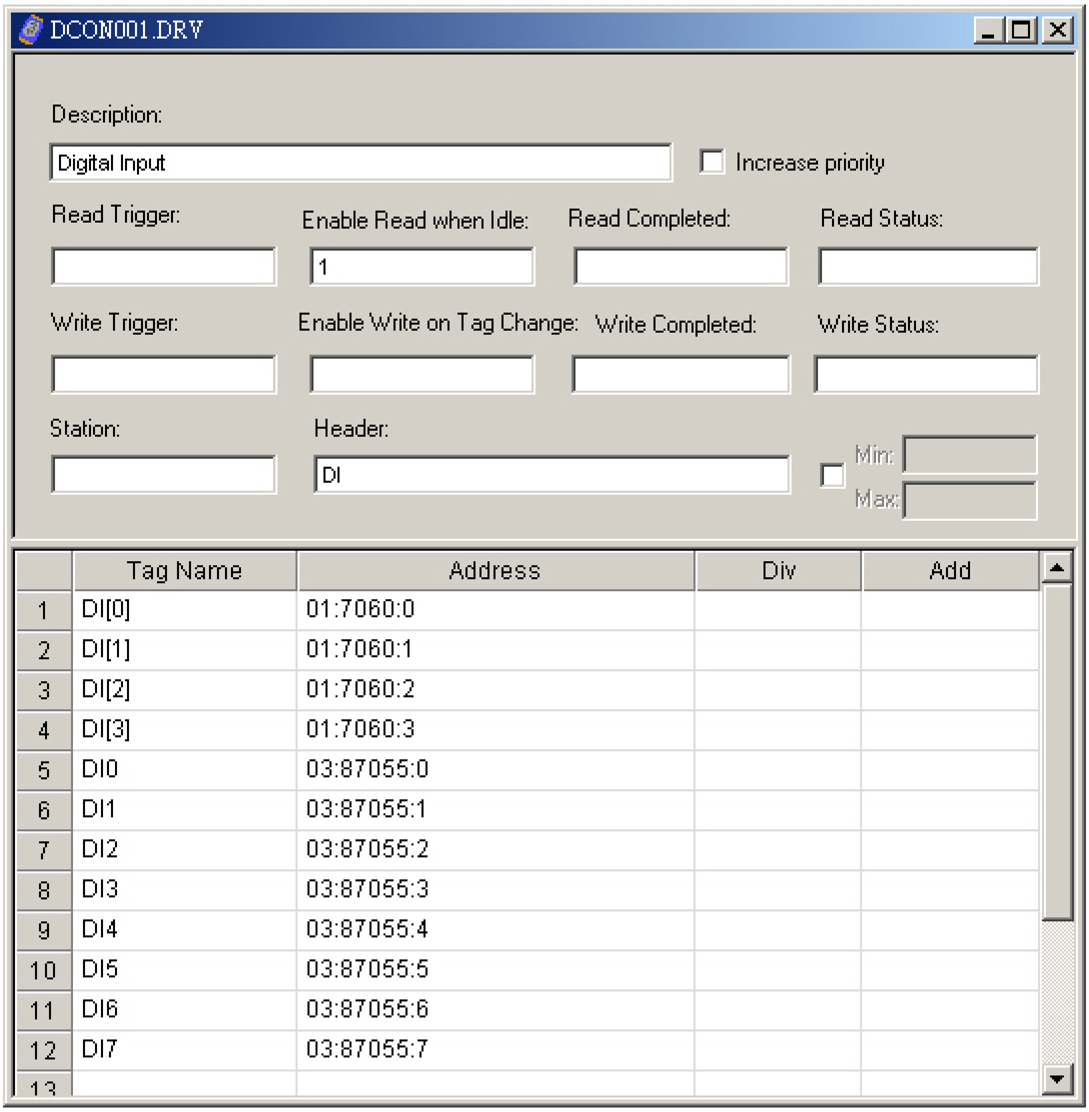
Station:

Header: DI

□ Min: Max:

	Tag Name	Address	Div	Add
1	DI[0]	01:7060:0		
2	DI[1]			
3	DI[2]			
4	DI[3]			
5	DI0	03:87055:0		
6	DI1			
7	DI2			
8	DI3			
9	DI4			
10	DI5			
11	DI6			
12	DI7			
13				

Figure 7-1 (**Fast Mode**)



Description: Digital Input

□ Increase priority

Read Trigger:

Enable Read when Idle: 1

Read Completed:

Read Status:

Write Trigger:

Enable Write on Tag Change:

Write Completed:

Write Status:

Station:

Header: DI

□ Min: Max:

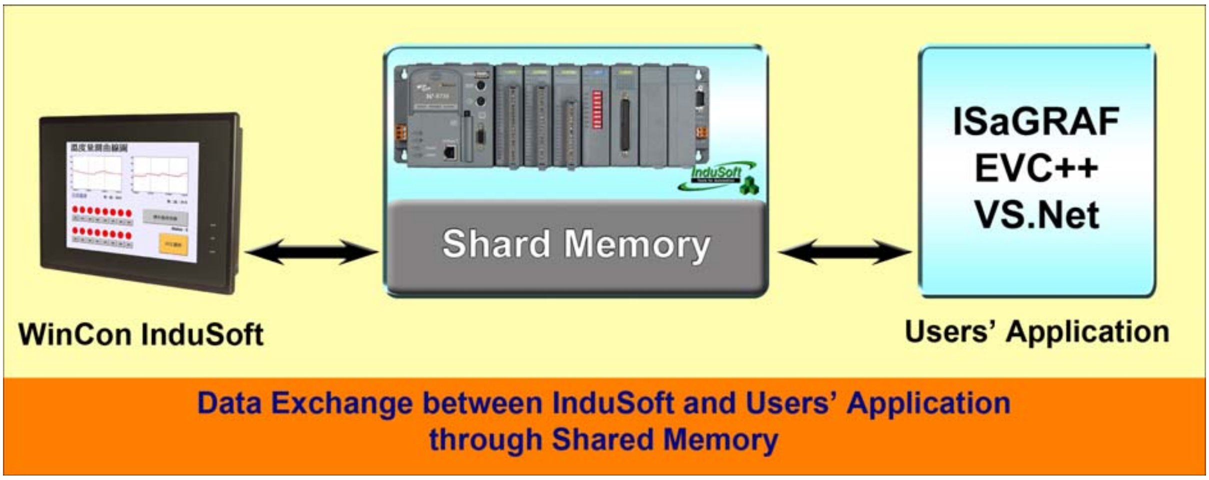
	Tag Name	Address	Div	Add
1	DI[0]	01:7060:0		
2	DI[1]	01:7060:1		
3	DI[2]	01:7060:2		
4	DI[3]	01:7060:3		
5	DI0	03:87055:0		
6	DI1	03:87055:1		
7	DI2	03:87055:2		
8	DI3	03:87055:3		
9	DI4	03:87055:4		
10	DI5	03:87055:5		
11	DI6	03:87055:6		
12	DI7	03:87055:7		
13				

Figure 7-2 (**Normal Mode**)

8. Data Exchange with Users' Application

The DCON Driver version 1.30 or above supports Quicker Shared Memory Function and the features are described as follows：

(1) It accomplishes that **Users' Applications (EVC++, VB.Net, C#, ISaGRAF) are able to exchange data with InduSoft.** Please refer to the following figure :



Data Exchange between InduSoft and Users' Application through Shared Memory

(2) The function is just used for indusoft running in the PAC platform not PC.

(3) The supported **Data Type** for communication between Users' Applications and InduSoft are：

"**Unsigned Char**"	(Range ： 0 ~ 255)
"**Short**"	(Range ： -32768 ~ 32767)
"**Long**"	(Range ： -2147483648 ~ 2147483647)
"**Float**"	(Range ： 3.4E +/- 38)

For Example :

If users want to exchange data between Users' Applications and InduSoft, please follow the steps :

(1) For Users' Applications of PAC, users must add "**UserShare Function**" provided by Quicker in the program that allows users to write or read data into shared memory. The program can be written by C#, VB.Net or EVC++.

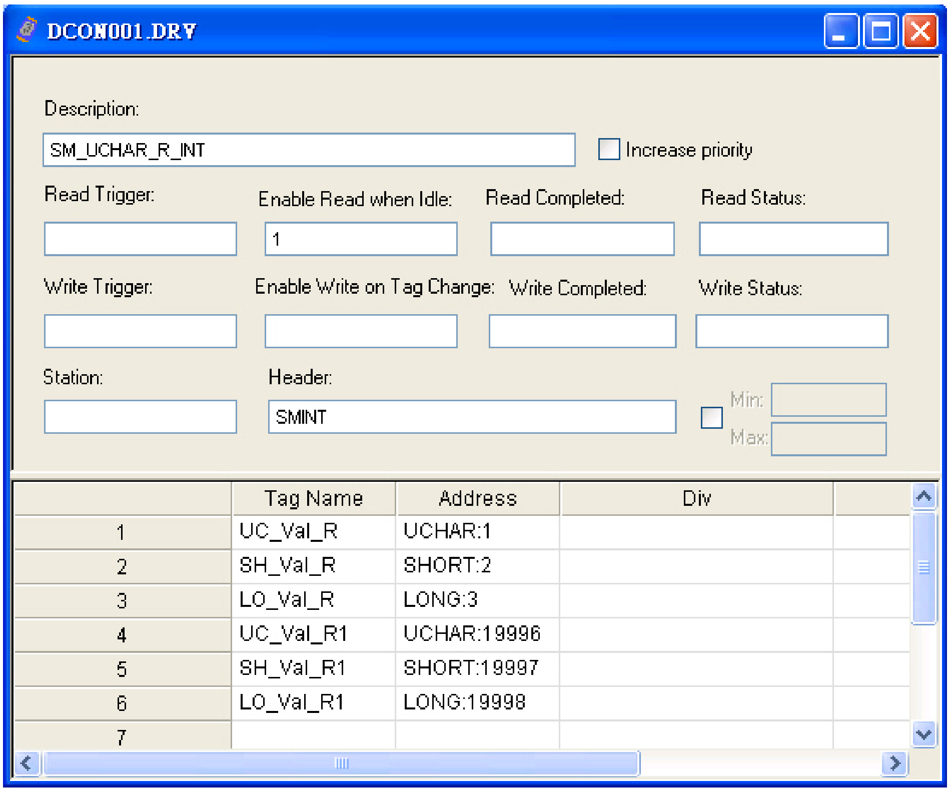
(2) For InduSoft Project, users just need to add a **DCON Driver Form** and fill it with the corresponding value. We provide four data type to Write/Read data into shared memory and the configuration refers to the following table.

Sample of Shared Memory Configuration		
Address on the Device	**Header Field**	**Address Field**
Write/Read "unsigned char" data into Shared Memory	**SMINT**	**UCHAR:1**
Write/Read "Short" data into Shared Memory	**SMINT**	**SHORT:2**
Write/Read "Long" data into Shared Memory	**SMINT**	**LONG:3**
Write/Read "Float" data into Shared Memory	**SMFLOAT**	**FLOAT:4**

The **Header Field** refer to the following syntax ：
SMINT ： When data type is "Unsigned Char", "Short", "Long"
SMFLOAT ： When data type is "Float".

The **Address Field** refer to the following syntax ：
<DataType> : <Address>
DataType ： "UCHAR", "SHORT", "LONG", "FLOAT"
Address ： The address which users want set assigned value in. (Range from **1** to **19999**)

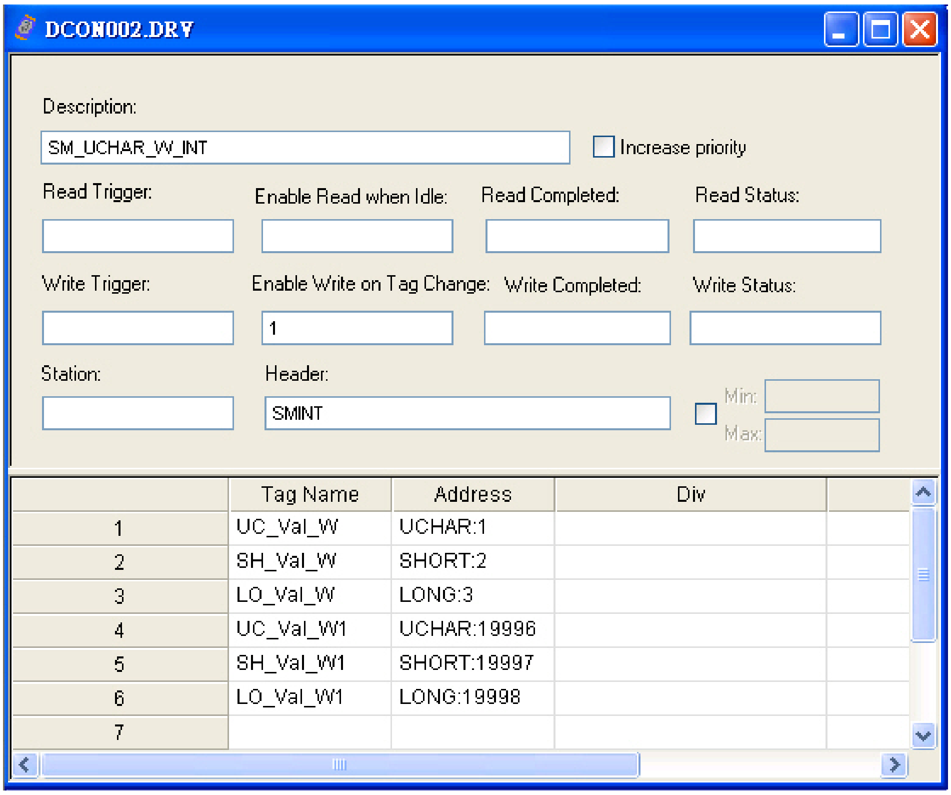
The following figures are the DCON Driver Form Demo about **how to Write/Read data into Shared Memory** in indusoft. The demo program (including Indusoft and EVC++ demo) for accessing Shared Memory can be downloaded in the path : http://www.icpdas.com/products/Software/InduSoft/indusoft.htm



Description: SM_UCHAR_R_INT
Enable Read when Idle: 1
Header: SMINT

	Tag Name	Address	Div
1	UC_Val_R	UCHAR:1	
2	SH_Val_R	SHORT:2	
3	LO_Val_R	LONG:3	
4	UC_Val_R1	UCHAR:19996	
5	SH_Val_R1	SHORT:19997	
6	LO_Val_R1	LONG:19998	
7			

Read data from Shared Memory For SMINT



DCON002.DRV

Description: SM_UCHAR_W_INT

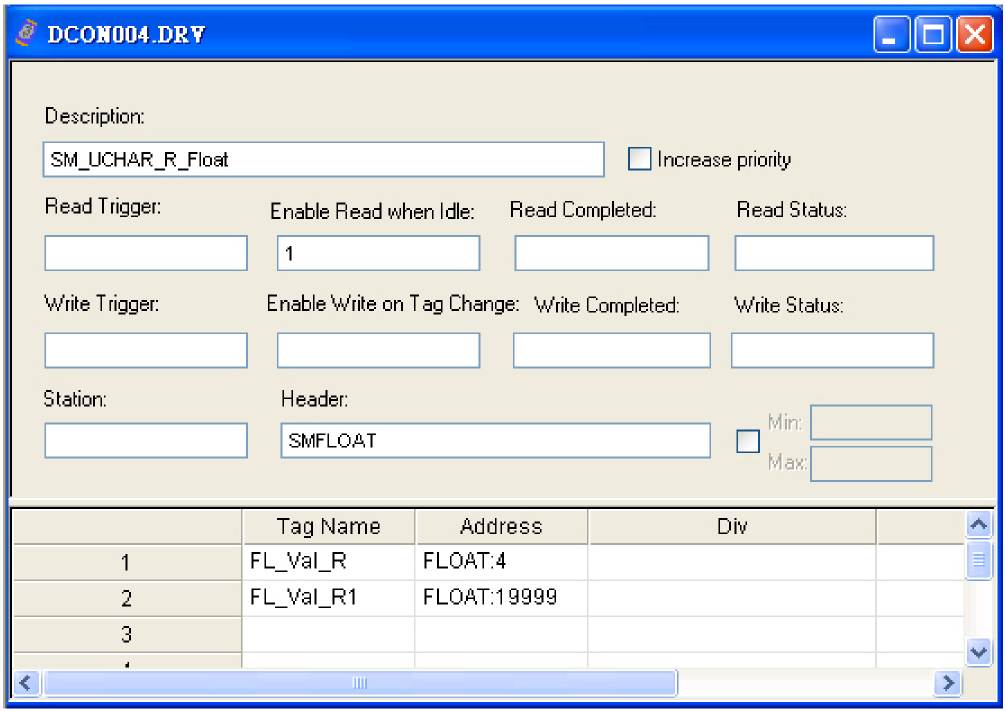
□ Increase priority

Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:
	1		
Station:	Header:		
	SMINT		

□ Min: Max:

	Tag Name	Address	Div
1	UC_Val_W	UCHAR:1	
2	SH_Val_W	SHORT:2	
3	LO_Val_W	LONG:3	
4	UC_Val_W1	UCHAR:19996	
5	SH_Val_W1	SHORT:19997	
6	LO_Val_W1	LONG:19998	
7			

Write data into Shared Memory For SMINT



DCON004.DRV

Description: SM_UCHAR_R_Float

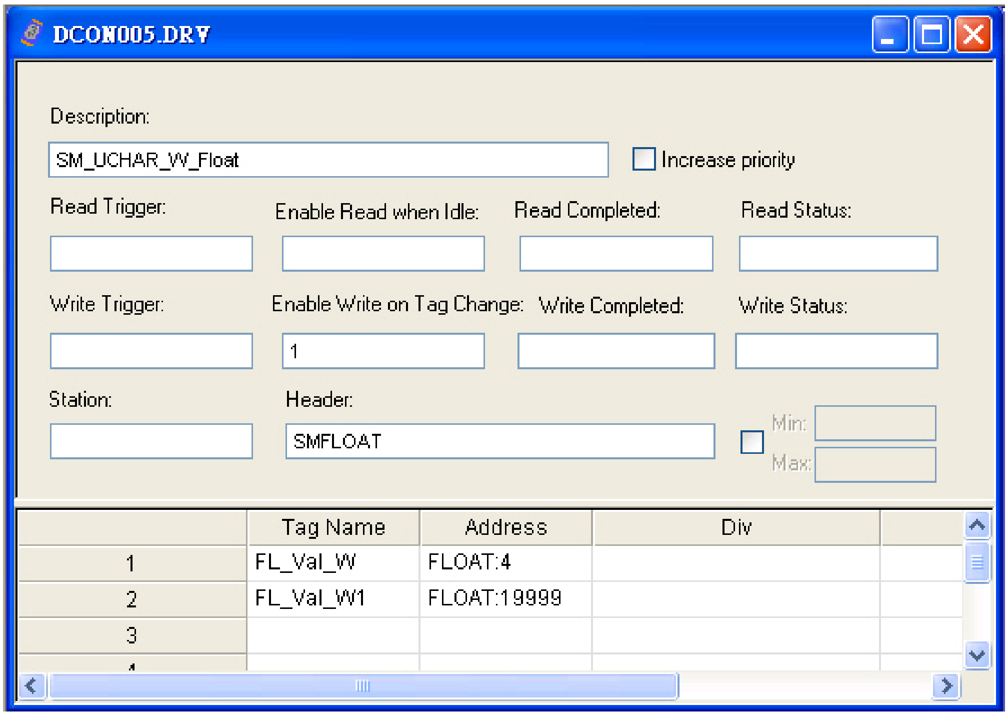
□ Increase priority

Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:
	1		
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:
Station:	Header:		
	SMFLOAT		

□ Min: Max:

	Tag Name	Address	Div
1	FL_Val_R	FLOAT:4	
2	FL_Val_R1	FLOAT:19999	
3			

Read data from Shared Memory For SMFLOAT



Description: SM_UCHAR_W_Float

□ Increase priority

Read Trigger:

Enable Read when Idle:

Read Completed:

Read Status:

Write Trigger:

Enable Write on Tag Change: 1

Write Completed:

Write Status:

Station:

Header: SMFLOAT

□ Min: Max:

	Tag Name	Address	Div
1	FL_Val_W	FLOAT:4	
2	FL_Val_W1	FLOAT:19999	
3			
4			

Write data into Shared Memory For SMFLOAT

9. Using Multi-DCON_Driver

When Users want to use two or more DCON Drivers in indusoft, please follow the below steps：

[1. For PC：]

Copy these three files -- "**DCON.dll**" , "**DCON.ini**" , "**DCON.msg**" located in the path "**C:\Program Files\Indusoft Web Studio v6.0\Drv**" and change the filename. Please follow these two rules for the filename of new DCON Driver.

(1) The total characters of new DCON Driver filename can't exceed five characters.

(2) The last character of new DCON Driver filename should be English word.

For example：

[1] DCONA, DCONB.... => these are right !!

[2] DCON1, DCON2, DCONAA => those are wrong !!

[2. For WinCE：]

The steps are all the same with PC. But the file path of DCON Driver for PAC is different and it is located in the "**\CompactFlahs\InduSoft\Drv**" of WinCon or "**\Micro_SD\InduSoft\Drv**" of WinPAC and ViewPAC.

[Note]

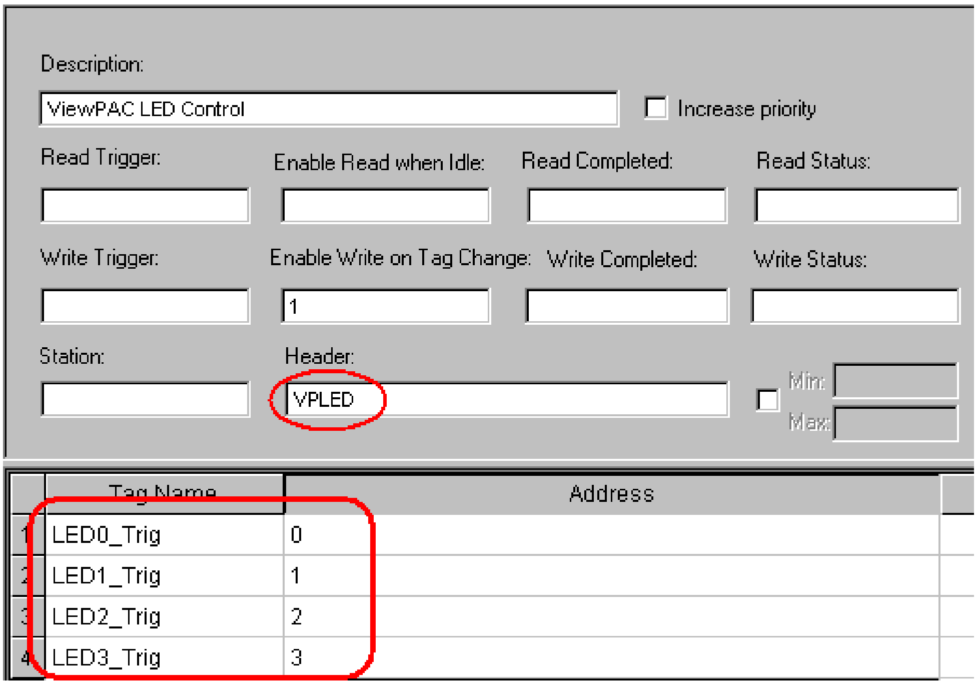
1. The com port number should be different in each DCON Driver.

10. ViewPAC LED/Buzzer Control Demo

10.1 ViewPAC LED Control

When Users want to control LED of ViewPAC controller, please follow the below steps：

(1) Fill "**VPLED**" in "Header" field of DCON Driver Form.
(2) In "Address" field, please input **0** or **1** or **2** or **3** and their meaning are as below.
0: **RUN** LED
1: **L1** LED
2: **L2** LED
3: **L3** LED
(3) In "Tag Name" field, please fill in a tag with Boolean type.
1: Turn on LED
0: Turn off LED



Description: ViewPAC LED Control
□ Increase priority

Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:

Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:
	1		

Station:
Header: VPLED
□ Min: Max:

	Tag Name	Address
1	LED0_Trig	0
2	LED1_Trig	1
3	LED2_Trig	2
4	LED3_Trig	3

10.2 ViewPAC Buzzer Control

When Users want to control buzzer of ViewPAC controller, please follow the below steps：

(1) Fill "**VPBUZZER**" in "Header" field of DCON Driver Form.
(2) In "Address" field, please input **0** or **1** or **2** or **3** and their meaning are as below.

0: **pac_BuzzerBeep** function	(For Write function)
1: **pac_StopBuzzer** function	(For Write function)
2: **pac_SetBuzzerFreqDuty** function	(For Write function)
3: **pac_GetBuzzerFreqDuty** function	(For Read function)

(3) In "Tag Name" field, please fill in a tag with **String** type.

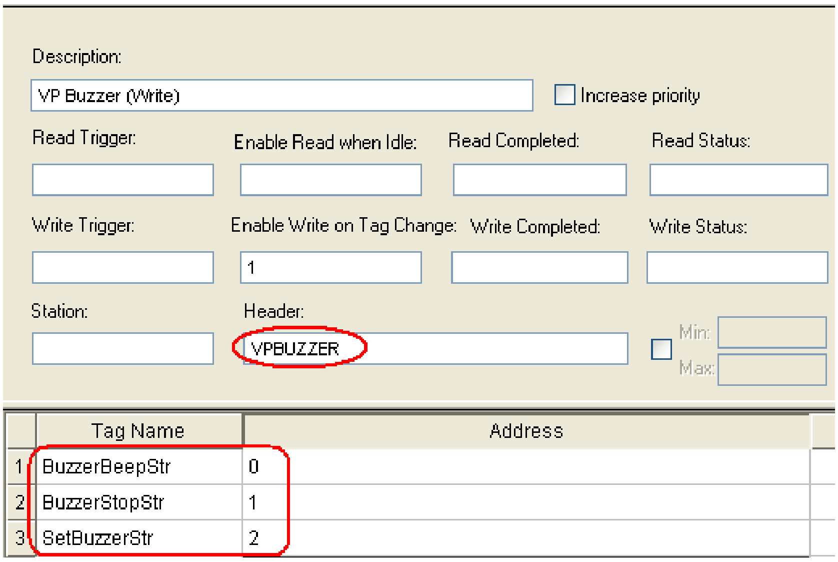
The string tag value has the following sequence:

<Param 1>:< Param 2>:< Param 3>: ・・・ < Param N>

Param 1 represents the first parameter of the API been called. The param 2 represents second parameter of the API and so on. Parameters are used with **Decimal** value.

[For Writing function of ViewPAC buzzer :]

Example of Tag Value	
Buzzer Function	**Tag Value (String Type)**
pac_BuzzerBeep (WORD count, DWORD milliseconds); **count:1 ; milliseconds:200**	**1:200**
pac_SetBuzzerFreqDuty (int freq, int duty); **freq:1000 ; duty:90**	**1000:90**



Description: VP Buzzer (Write) □ Increase priority

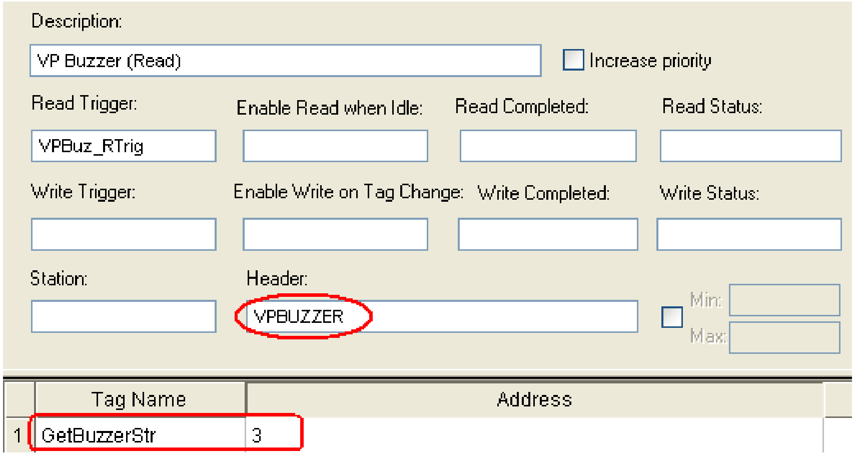
Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:
Write Trigger:	**Enable Write on Tag Change:**	**Write Completed:**	**Write Status:**
	1		

Station:
Header: VPBUZZER □ Min: Max:

	Tag Name	Address
1	BuzzerBeepStr	0
2	BuzzerStopStr	1
3	SetBuzzerStr	2

[For Reading function of ViewPAC buzzer :]

Example of Tag Value	
Buzzer Function	**Tag Value (String Type)**
pac_GetBuzzerFreqDuty(int *freq, int *duty)	**1000:90** **(Reading => freq:1000 ; duty:90)**



Description: VP Buzzer (Read) □ Increase priority

Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:
VPBuz_RTrig			
Write Trigger:	**Enable Write on Tag Change:**	**Write Completed:**	**Write Status:**

Station:
Header: VPBUZZER □ Min: Max:

	Tag Name	Address
1	GetBuzzerStr	3

11. Troubleshooting

After each attempt to communicate using this driver, the tag configured in the field **Read Status or Write Status** will receive the error code regarding the kind of failure that occurred. The error messages are:

Error Code	Description	Possible causes	Procedure to solve
0	NoError	Communication without problems.	
1	Invalid Header	An invalid Header has been typed or the tag that is inside this field has an invalid configuration.	Type a valid Header either on the header field or on the tag value. A lot of different valid headers are shown on the section 4.2
2	Invalid Address	An invalid Address has been typed or the tag that is inside this field has an invalid configuration.	Type a valid Address either on the address field or on the tag value. The address' valid values are show on the section 4.4.
10	SendCmdError	Send command error.	Check the serial communication Configuration. Verify if the settings on the Communication Parameters and on the device are the same.
12	ResultStrCheckError	Result string check error.	- Check the cable wiring - Check the PLC state. It must be RUN - Check the station number. - Check the right configuration.
15	TimeOut	Time Out error.	- Check the cable wiring - Check the PLC state. It must RUN - Check the station number. - Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.
17	ModuleIdError	Module ID error.	-Check supports the Module. See on the section 4.2.
18	AdChannelError	Channel number error.	-Check the AI channel number.
19	UnderInputRange	Under input range error.	-Check the input range. See if the value is valid.
20	ExceedInputRange	Exceed input range error.	-Check if the input value is valid.
21	InvalidateCounterNo	Invalid counter number.	-Check if the counter number value is valid.
22	InvalidateCounterValue	Invalid counter value.	-Check if the counter number value is valid.
25 / **65533**	Open Com Fail	Invalid com port number / com port is in use.	-Check if com port number is valid. -Check if com port is in use.

Note: The results of the communication may be verified in the **output** Window of the Studio's environment. To set a log of events for **Field Read Commands**, **Field Write Commands** and **Serial Communication** click with the right button of the mouse on the output window and choose the option setting to select these log events.

When testing under a Windows CE target, you can enable the log at the unit (Tools/Logwin) and verify the file celog.txt created at the target unit.

When you are not able to establish the communication with the PLC, you should first of all establish the communication between the PLC Programming Tool and the PLC. Very frequently the communication it is not possible due to a hardware or cable problem, or due an error or lack of configuration at the PLC. Only after the communication between the PLC Programming Software and the PLC is working fine, you can test again the supervisory. When testing the communication with the Studio, you should first use the application sample described at item 7, instead of the new application that you are creating.

If is required to contact technical support, please have the following information available:
- Operating System (type and version): To find this information use the Tools/System Information option
- Project information: It is displayed using the option Project/Status from the Studio menu
- Driver version and communication log: Available from Studio Output when running the driver
- Device model and boards: please refer to hardware manufacture's documentation

12. Application Sample

The Studio contains a configured project to test the driver. It is strongly recommended to do some tests with this application before beginning the configuration of the customized project, for the following reasons:

- **To understand better the information covered in section 4 of this document.**
- **To verify that your configuration is working.**
- **To certify that the hardware used in the test (device + adapter + cable + PC) is in working conditions before beginning the configuration of the applications.**

2 Note: The Application Sample is not available for all drivers.

The Studio application is in the directory:

To perform the test, you need to follow these steps:

- **Configure the device communication parameters using manufacture programmer software.**
- **Open the application**
- **Execute the application**
- **To display the following screen with some information about the communication, please execute the Viewer module in the Studio.**

3 Note: The Application for testing may be used like a maintenance screen for the custom application.

13. History of Versions

Version	By	Date	Description of changes
1.01	Anold	15-Sep-02	First driver version
1.02	Troy	30-Sep-03	Include driver for W-8x3x
1.05	Troy	05-May-04	1. Fixed I-8017 channel numbers 2. Supports W-803x
1.10	Troy	07-Dec-04	1. Supports new DCON series modules 2. Add **timeout** and **checksum** settings for Communications Parameters 3. To **speed up** the AI/DI channel scan 4. Choose I-8017/24 Input/Output range in W-8x3x
1.20	Edward	15-Aug-05	1. Supports new DCON series modules 2. ”**SendCmd**” Function supports **87k** modules in WinCon slot
1.30	Edward	28-Dec-05	1. Supports “**Data Exchange with Users’ Application**” Function
1.31	Edward	15-Feb-06	1. Support **I-7005**
1.32	Edward	21-Jun-06	1. Support from **“Read_Status”** field of InduSoft DCON Driver Form to know the module communication status
1.33	Edward	26-Mar-07	1. Support AI Modules “**Break Line Detection**” function.
1.34	Edward	18-May-07	1. Correct the return value of “**Write Status**” field in InduSoft DCON Driver Form when write data failed. 2. Support AI module – “**I-87018Z**” (10 AI channel)
1.35	Edward	03-Mar-08	[NT] 1. Correct I-7060 DO Output problem. 2. Correct I-7016 read AI problem. [CE] 1. Support WinPAC platform. 2. Correct I-87K read AI problem.
1.36	Edward	08-April-08	[NT] 1. Support Multi-DCON Driver. [CE] 1. Support Multi-DCON Driver. 2. Correct I-7055 DO Output problem
1.37	Edward	26-May-08	[NT] 1. Fix AI module reading problem when using DO modules. (for i-7k modules)

1.38	Edward	17-Feb-09	[NT][CE] 1. Support I-7013 / I-8046 / I-8048 / I-87005 / I-87022 [CE] 1. Fix AO:4 Current Output problem.
1.39	Edward	31-Aug-09	[CE] 1. Support I-8094 / I-8094Fmodule. 2. Add ViewPAC L1~L3 LED control function. 3. Fix WinPAC/ViewPAC Slot-0 bug.
1.40	Edward	30-Nov-09	[NT][CE] 1. Support I-7017 up to 20 channels. 2. Fix the error for reading data of the last column configured in a driver sheet [CE] 1. Modified the I-8094/I-8094F communication methods.
1.41	Edward	29-Mar-10	[NT] 1. Modify DO output error when using multi DO modules
1.42	Edward	03-May-10	[CE] 1. Add ViewPAC Buzzer control function.